

REMARKS/ARGUMENTS

The Examiner rejected 1-9, 11-19, and 21-19 as obvious (35 U.S.C. §103) over Davidson (U.S. Patent No. 6,083,276). Applicant traverses the rejections for the following reasons.

Independent claims 1, 11, and 21 concern generating an interface to elements in a document, wherein the document defines a relationship of the elements and at least one attribute for each element. These claims require generating a class implementing methods for at least one element from information provided on elements in the document and a mapping indicating at least one element in the document to map to a class, wherein the at least one indicated element in the document for which the class is generated can be accessed and affected by the methods implemented in the class.

The Examiner cited col. 24, lines 50-63 as teaching the claim requirement of generating a class implementing methods for at least one element in the document from information provided in the document and mapping an element to a class. The cited col. 24 discusses how an element processor 118 expects the target component 212 to provide a description of methods on the target component. The element processor 118 maps attributes of the element in the document to properties in the component 212. (Col. 24, lines 35-41)

Nowhere does the cited col. 24 anywhere teach or suggest the claim requirement of generating a class of methods for one element in the document. Instead, the cited col. 24 discusses how a component is generated from the elements. The claims, on the other hand, require a class implementing methods used to access one element in the document. The cited col. 24 provides methods for use on a component, not the elements in the document as claimed.

The Examiner cited col. 23, line 4 to col. 24, line 10 and 50 to col. 25, line 27 of Davidson as teaching the claim requirement that the at least one indicated element in the document for which the class is generated can be accessed and affected by the methods implemented in the class. (Office Action, pg. 3) Applicant traverses.

The cited col. 23, line 40-55 discusses how an element processor 118 creates an instance of a target class 132 associated with an element tag. The element processor 118 then creates the component 212 that is an instance of the class 132. After the component 212 is instantiated, the element processor 118 maps attributes from the document to the component 212. (Col. 24, lines 28-35) As discussed, the cited Davidson discusses how to create a component having elements

from the document and classes used with that component. However, nowhere does the cited Davidson anywhere teach or suggest the claim requirement of generating methods that can access and affect the element in the document. Instead, the cited Davidson discusses a class 132 created to use a component 212, not the elements in the document as claimed.

The cited col. 25 discusses a method that can be used to set a property value in the component 212. Again, this cited Davidson discusses methods to act on the component 212, which is separate from the file 202 having the elements. See, FIG. 2 of Davidson. Nowhere does the cited Davidson anywhere teach or suggest the claim requirement of generating methods that can access and affect elements in the document. This distinction is further emphasized in the “Summary of the Invention” of Davidson which states that the components are used to launch a component based application. (Col. 4, lines 48-62) Nowhere does Davidson discuss processing the document and mapping to generate methods used to access and affect elements in the document.

The Examiner found that Davidson on col. 6, lines 59-65 and col. 7, lines 61-67 teaches that the component is of the target class which maps to an element. (Office Action, pg. 3) The cited cols. 6 and 7 discuss mapping elements to components. An element mapper used to select an element processor. An element mapper maps the element’s name to a class, and then instantiates the component from that class. The element processor uses an attribute mapper to map attributes of the elements to the component. Although Davidson does discuss mapping an element to a class, and then instantiating the component from the class to have attributes mapped from the element, nowhere does the cited Davidson anywhere teach or suggest generating methods that are used to access and effect the element in the document. Instead, Davidson discusses accessing properties in a component 212 that is separate from the file 202 defining the elements. See, FIG. 2 of Davidson. Davidson thus provides access to element properties in a component, not an interface to elements in the document as claimed.

For all the above reasons, Applicant submits that claims 1, 11, and 21 are patentable over the cited art because the cited Davidson does not teach or suggest all the claim requirements.

Dependent claims 2-10, 12-20, and 22-30 are patentable over the cited art because they depend from claims 1, 11, and 21, which are patentable over the cited art for the reasons

discussed above. Moreover, the below discussed dependent claims provide additional grounds of patentability over the cited art.

Claims 2, 12, and 22 depend from claims 1, 11, and 21 and further require that the mapping includes a class name for each indicated element. The Examiner cited col. 7, lines 61-67 of Davidson as teaching the additional requirements of these claims. (Office Action, pg. 4) Applicant traverses.

The cited col. 7 mentions that the element mapper maps an element name to a corresponding target class, from which the component is instantiated. Nowhere does the cited col. 7 teach or suggest a mapping that includes a class name for an element. Instead, the cited col. 7 discusses mapping an element name to a class, not providing a class name for an element as claimed.

Accordingly, claims 2, 12, and 22 provide additional grounds of patentability over the cited art.

Claims 4, 14, and 24 depend from claims 1, 11, and 21 and further require that the relationship of the elements in the document are arranged in a hierarchical relationship, and wherein the methods in the at least one class generated for the element allow a user to directly access and affect the element in the document. Applicant amended these claims to clarify that the class allows the user to access and affect the element in the document.

The Examiner cited col. 7, lines 6-47 as teaching the additional requirements of claims 4, 14, and 24. (Office Action, pg. 4) Applicant traverses for the following reasons.

The cited col. 7 discusses how the ADF document is parsed into a parse tree having the elements and attributes. The parse tree is then transformed into components 212. Applicant submits that nowhere does the cited col. 7 teach or suggest generating methods to allow a user to directly access and affect the element in the document. Instead, Davidson generates a component 212 having information from the file defining the element that may be accessed.

Accordingly, claims 4, 14, and 24 provide additional grounds of patentability over the cited art.

Claims 5, 15, and 25 depend from claims 4, 14, and 24 and additionally require accessing the at least one element in the document indicated in the mapping using a hierarchical application program interface (API), wherein one class is generated for each accessed element. The

Examiner cited col. 6, lines 27-37 of Davidson as teaching the additional requirements of these claims. (Office Action, pg. 4). Applicant traverses.

The cited col. 6 discusses an application framework that has a plurality of classes. Although the cited col. 6 mentions classes related to an element, nowhere does the cited col. 6 anywhere teach or suggest accessing an element in the document using a hierarchical application program interface, where there is one class for each element. Instead, the cited col. 6 discusses classes in general.

Accordingly, claims 5, 15, and 25 provide additional grounds of patentability over the cited art.

Claim 6, 16, and 26 depend from claims 1, 11, and 21, respectively, and further require that the mapping indicates an interface to generate for the class, wherein the interface defines methods to access the element for which the class is generated. The Examiner cited the previously discussed cols. 6, 7, and 24 as teaching the above claim requirements. (Office Action, pg. 4) As discussed, the cited cols. 6, 7, and 24 nowhere teaches or suggests generating interfaces to access the element itself. Instead, the cited Davidson discusses methods to access a component 212 generated from a parse 204 tree that is generated from a file 202 defining the element. See, FIG. 2 of Davidson.

Accordingly, claims 6, 16, and 26 provide additional grounds of patentability over the cited art.

Claims 7, 17, and 27 depend from claims 6, 16, and 26 and further require that the methods implemented in the class include at least one method that is a member of the set of methods comprising: adding an instance of the element, inserting an instance of the element at a location in the document with respect to other instances of the element in the document, and removing an instance of the element from the document. The Examiner cited col. 23, line 4 to col. 24, line 10 as teaching the additional requirements of these claims. (Office Action, pg. 5) Applicant traverses.

The cited cols. 23 and 24 discuss the element processor that creates a target class associated with an element tag, which is then used to instantiate the component having attributes from the element. Nowhere do the cited cols. 23 and 24 anywhere teach or suggest methods that insert, remove or add an instance of the element in the document as claimed. Instead, the cited

Davidson discusses methods to access the component 212, not elements in the document as claimed.

Accordingly, claims 7, 17, and 27 provide additional grounds of patentability over the cited art.

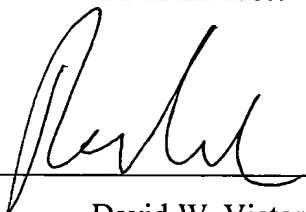
Claims 8-10, 18-20, and 28-30 are patentable over the cited art because they depend from claims 1, 11, and 21, which are patentable over the cited art for the reasons discussed above, and because their additional requirements in combination with the base and any intervening claims provide further distinction over the cited art. Moreover, the Examiner rejected claims 10, 20, and 30 as obvious in view of Davidson and another patent, Skinner (U.S. Patent No. 6,085,198). However, Skinner was applied for the additional requirements of claims 10, 20, and 30, not the requirements of the base claims 1, 11, and 21 which in combination with the dependent claims provide still further grounds of patentability over the cited art.

Conclusion

For all the above reasons, Applicant submits that the pending claims 1-30 are patentable over the art of record. Applicant submits herewith the fee for a One Month Extension of Time. Nonetheless, should any additional fees be required, please charge Deposit Account No. 09-0460.

The attorney of record invites the Examiner to contact him at (310) 553-7977 if the Examiner believes such contact would advance the prosecution of the case.

Dated: July 11, 2003

By: 
David W. Victor
Registration No. 39,867

Please direct all correspondences to:

David Victor
Konrad Raynes Victor & Mann, LLP
315 South Beverly Drive, Ste. 210
Beverly Hills, CA 90212
Tel: 310-553-7977
Fax: 310-556-7984